iSLS9 data analysis workshop, Part II

Hyungwon Choi

Department of Medicine, NUSmed

March 2, 2021

Contents

1 Data management and inspection **1**

2 Dimension reduction and sample projection analysis **8**

3 Understanding the correlation structure between lipids and other risk factors **15**

4 Association analysis: logistic regression **18**

5 Association analysis with Cox regression (time-to-event) **24**

1 Data management and inspection

In this section, we introduce basic operations in R language to inspect, clean up, and work with the data frame (spreadsheet). We will familiarize ourselves with command line syntax of R, and this process will get you ready for subsequent data analysis steps.

We will first read the QC'ed lipidomics data and the sample meta data into the workspace.

```
#getwd() ## verify your current working directory
ldata = read.delim("lipid_data.txt", header=T, as.is=T, check.names=F)
sdata = read.delim("sample_data.txt", header=T, as.is=T, check.names=F)

colnames(ldata) = gsub("\\(-H2O\\)", "", colnames(ldata))
### Remove the neutral loss tag in the lipid names
```

```
ldata[1:5,1:5]
```

```
##   ID Cer d16:1/C16:0 Cer d16:1/C18:0 Cer d16:1/C20:0 Cer d16:1/C22:0
## 1 S1    0.0018321806     0.002185681     0.006609402      0.02718963
## 2 S2    0.0013672951     0.001605267     0.003260313      0.02328471
## 3 S3    0.0013615316     0.002492004     0.006519285      0.03365524
## 4 S4    0.0008583807     0.003512810     0.005867816      0.01619162
## 5 S5    0.0017791874     0.002861309     0.007761635      0.02653510
```

```
sdata[1:5,]
```

```
##   ID DM Days Age Gender   BMI HbA1c SBP  HDL  LDL  CRP   TG Insulin Glucose
## 1 S1  0   NA  52      2 20.45    NA 105 1.76 4.16 0.29 1.12    4.65    4.43
## 2 S2  0   NA  54      1 22.93 5.558 137 1.64 2.74 0.72 0.90    5.87    4.30
## 3 S3  0   NA  52      1 24.47 5.267 111 1.44 2.91 0.26 2.68    6.78    4.24
## 4 S4  0   NA  53      1 24.21 6.062 143 1.71 2.36 1.30 1.20    3.96    5.22
## 5 S5  1 2190  65      2 29.23 4.987 150 1.63 2.95 3.85 1.65   12.35    5.72
```

As the output of the last two commands shows, most data are organized in spreadsheets. In mathematics we call spreadsheet a "matrix", which can be considered as stacks of column vectors (vertical) and row vectors (horizontal). In this case, we have saved both the sample meta data and the lipidomics data in a way that column vectors represent one set of information (e.g. a lipid, or a clinical parameter), whereas study participants are listed along the rows.

Now that we have the data read into the workspace, we check whether the sample meta data and the lipidomics data are properly aligned.

```
### First match the sample IDs between the two data sets
all(ldata$ID %in% sdata$ID)
```

```
## [1] TRUE
```

```
all(sdata$ID %in% ldata$ID)
```

```
## [1] TRUE
```

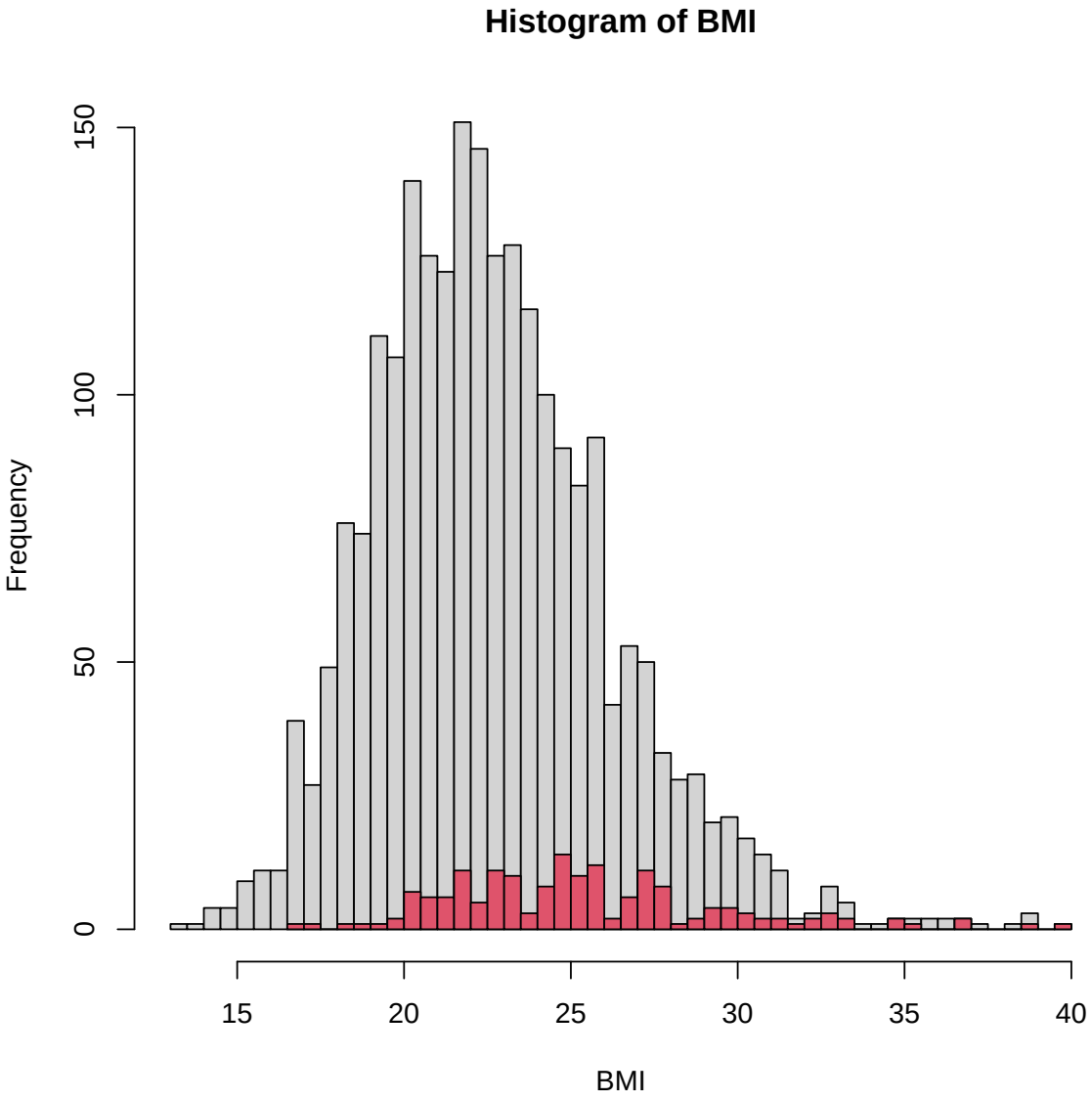
```
all(ldata$ID == sdata$ID) ## All sample IDs are aligned between the two data sets
```

```
## [1] TRUE
```

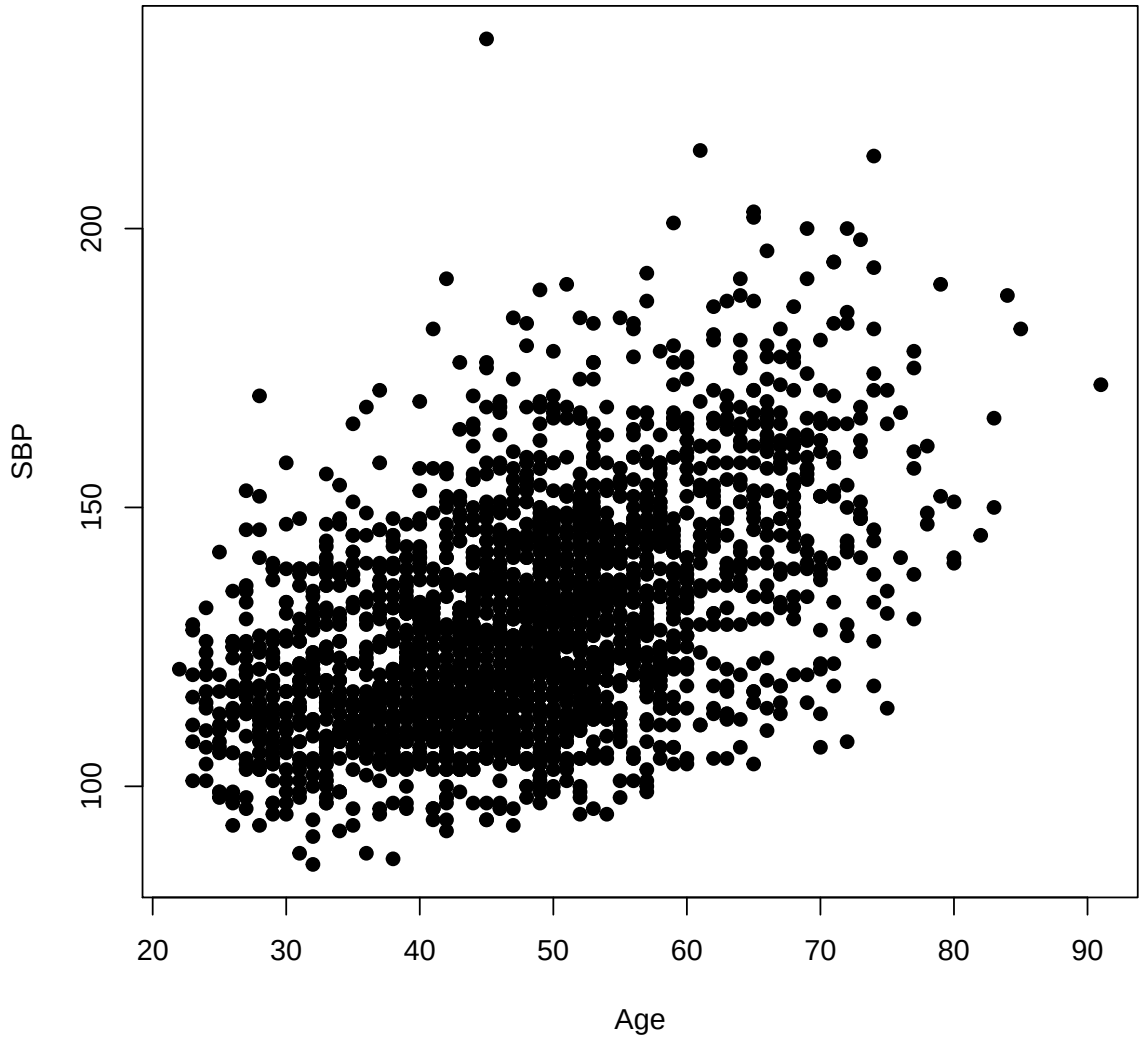
Now we are going to start exploring the data by quick visualization. In this section, we will "attach"

the data frame first, so we can call the variable names easily. After we are done with plotting the data, we will detach the data frame to avoid confusion in the rest of the tutorial

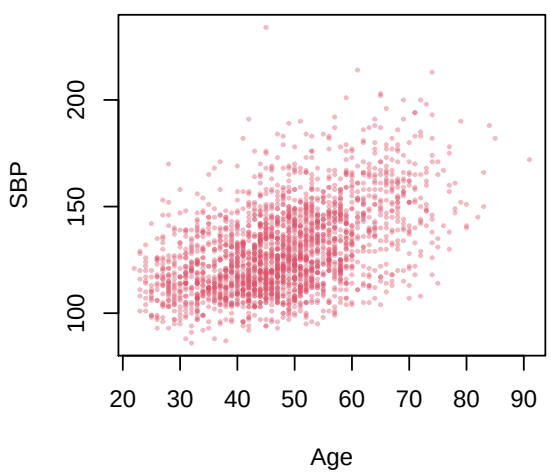
```
#################################
### Plotting sample meta data
### while learning R syntax
#################################

#install.packages("scales")
library(scales)

attach(sdata)
nsamples = nrow(sdata)

hist(BMI, breaks=50)
hist(BMI[DM == 1], breaks=50, add=TRUE, col=2)
```

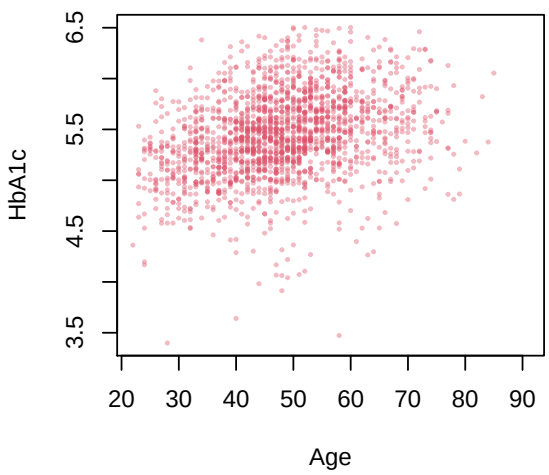


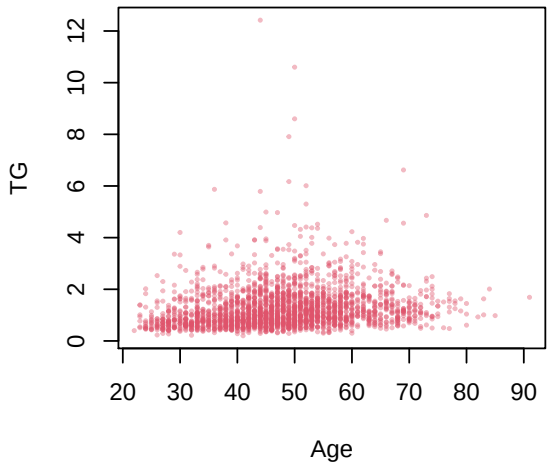

```
plot(SBP ~ Age, pch=19)
```

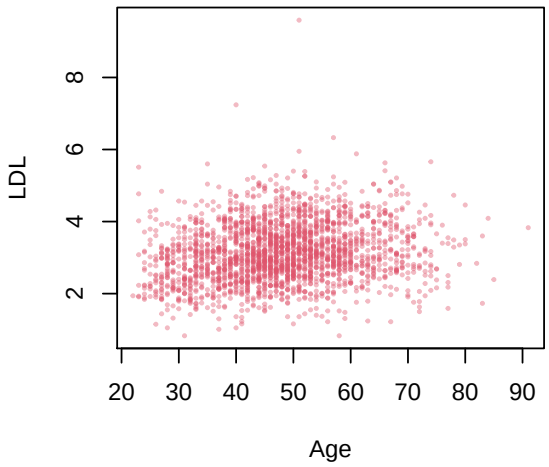



```
plot(SBP ~ Age, cex=.3, pch=19, col=alpha(2, 0.4))  ### red, opaque dots


par(mfrow=c(2,2))
plot(SBP ~ Age, cex=.3, pch=19, col=alpha(2, 0.4))
plot(HbA1c ~ Age, cex=.3, pch=19, col=alpha(2, 0.4))
plot(TG ~ Age, cex=.3, pch=19, col=alpha(2, 0.4))
plot(LDL ~ Age, cex=.3, pch=19, col=alpha(2, 0.4))
```







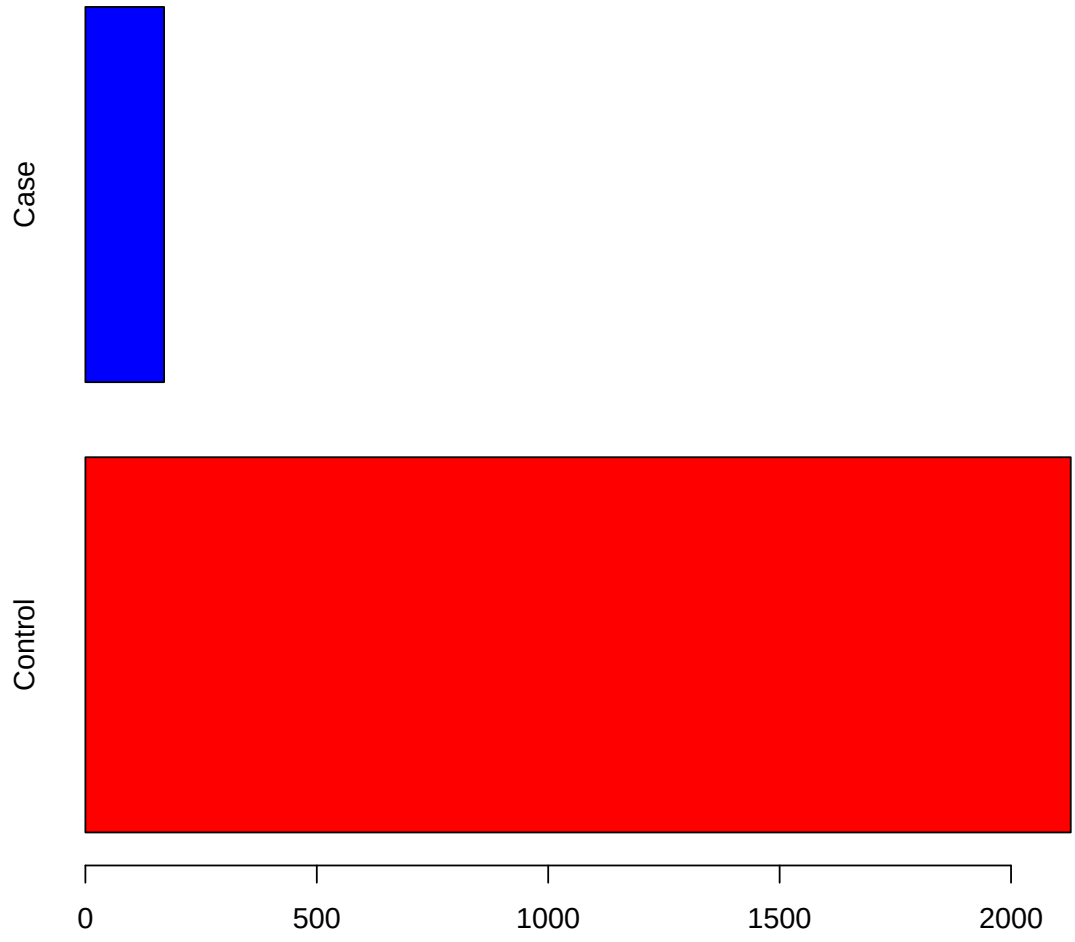



```
dev.off()

## null device
##           1

## Boxplots
par(mfrow=c(1,2))
boxplot(BMI ~ Gender, boxwex=0.3)
boxplot(HbA1c ~ DM, boxwex=0.3)
dev.off()
```

```
## null device
##           1
```

So far the plotting exercise involved continous variables only. How about categorical variables?

```
barplot(table(DM), width=0.2, col=c("red","blue"), horiz=TRUE, names.arg = c("Control","Case"))
```



Optionally, we can also test if there is any association between two categorial variables.

```r
## Chi-squared test for categorical data
## and see if there is any correlation / association
tab = table(DM, Gender)
print(tab)
```

```
##    Gender
## DM    1    2
##   0  977 1152
##   1   78   92
```

```r
chisq.test(tab)  ### not significant
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tab
## X-squared = 1.9136e-29, df = 1, p-value = 1
```

```r
## Discretizing a continuously scaled variable
bmi.new = rep(NA, nsamples)
bmi.new[BMI <= 18.5] = 1
bmi.new[BMI > 18.5 & BMI <= 23] = 2
bmi.new[BMI > 23 & BMI <= 27.5] = 3
bmi.new[BMI > 27.5] = 4
tab = table(DM, bmi.new)
print(tab)
```

```
##    bmi.new
## DM    1    2    3    4
##   0  229 1054  678  168
##   1    3   50   76   41
```

```r
chisq.test(tab)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab
## X-squared = 79.386, df = 3, p-value < 2.2e-16
```

```
### significant
detach(sdata)
```

2 Dimension reduction and sample projection analysis

In this section, we will start working with the lipidomics data. Before going into the unsupervised projection analysis, there are a few things we need to do:

```
rownames(ldata) = ldata$ID ### First column of the data matrix contains identifiers
ldata = ldata[,-1]
### Since the downstream data analysis will be performed on numerical data only,
### we put them into ``rownames'' and exclude them from the analysis.

### Check zero or negative concentrations before log transform
any(ldata <= 0)

## [1] TRUE

sum(ldata <= 0)

## [1] 1

### turns out only one value

### In case there are zeros,
### we plug-in a minimum value for each analyte.
### To accomplish this, we run through a "for" loop.
for(k in 1:ncol(ldata)) {
  xvec = ldata[,k]  ### temporary storage
  xpos = xvec[xvec > 0]  ### subset positive values
  ximpute = 0.9 * min(xpos)  ### get the 90% of the minimum intensity
  zid = (xvec <= 0)  ### TRUE/FALSE vector
  if(any(zid)) {  ### Perform this only if there is zero
    cat(colnames(ldata)[k])  ### 'cat' = print out
    xvec[zid] = ximpute  ### Plug in values wherever there is zero
    ldata[,k] = xvec  ### Store back to the original data frame
```

```
  }
}

## SM d18:1/C23:1

tmp = log2(ldata)
```

Frequency distribution of most ion count or sequence read count data is skewed, i.e. their distribution is often lopsided. Many statistical anaylsis methods implicitly assume that data are bell-shaped (normally distributed). To avoid conflict, it is often preferred to transform data to meet the assumption.

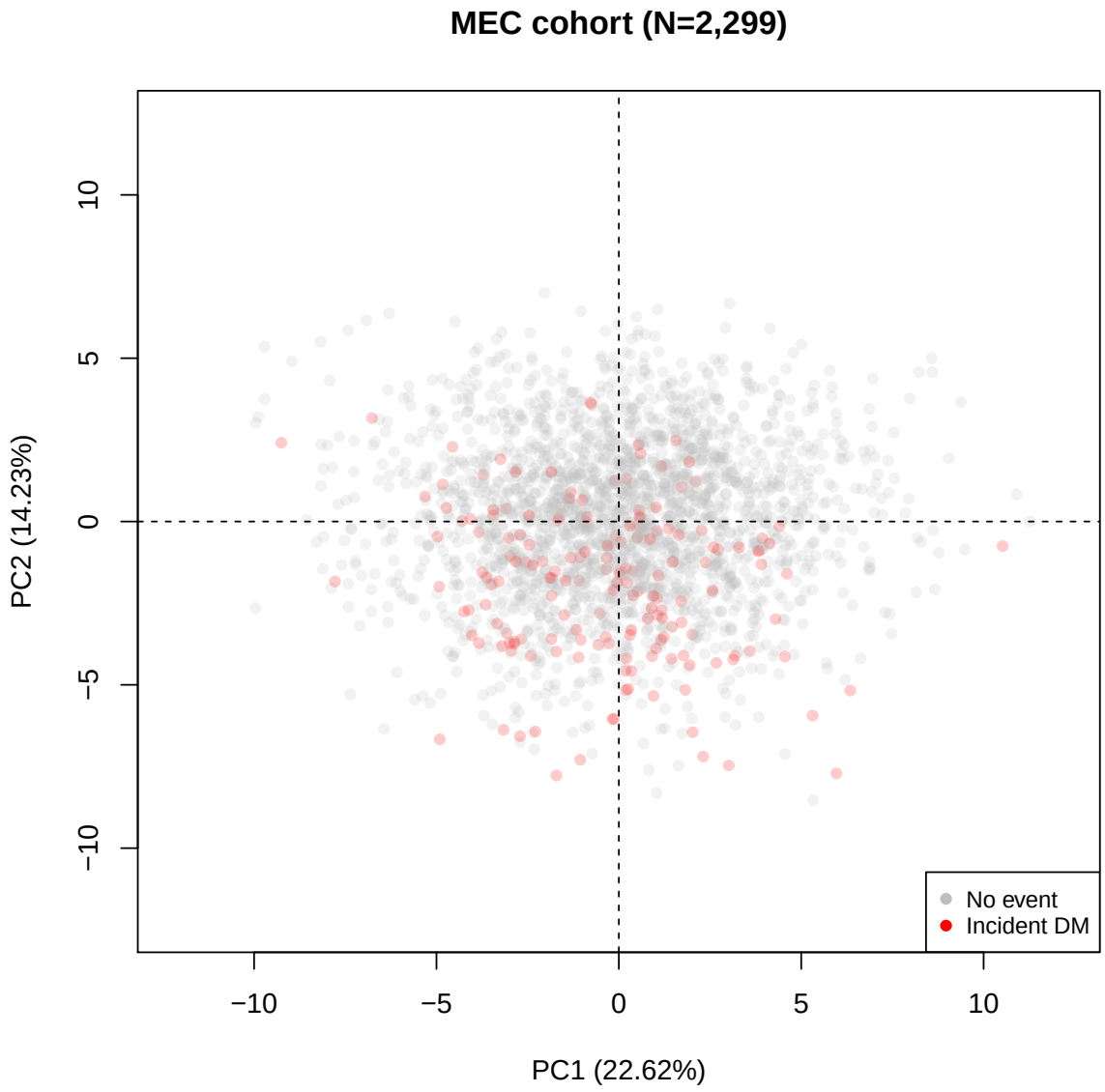
Now we try principal component analysis (PCA):

```
tmp.pca = prcomp(tmp)
vv = tmp.pca$sdev^2
vv = vv / sum(vv) * 100
vv = round(vv, 2)
print(vv)

##   [1] 22.62 14.23  6.14  5.21  3.91  3.70  3.10  2.49  2.25  1.99  1.87  1.79
##  [13]  1.55  1.32  1.17  1.16  1.03  0.97  0.89  0.83  0.78  0.72  0.67  0.61
##  [25]  0.59  0.58  0.54  0.53  0.52  0.48  0.46  0.42  0.41  0.39  0.38  0.37
##  [37]  0.34  0.34  0.33  0.32  0.31  0.30  0.29  0.28  0.27  0.27  0.26  0.25
##  [49]  0.24  0.24  0.23  0.23  0.23  0.21  0.21  0.20  0.20  0.20  0.20  0.19
##  [61]  0.19  0.18  0.18  0.17  0.17  0.17  0.16  0.16  0.16  0.16  0.15  0.15
##  [73]  0.15  0.15  0.14  0.14  0.14  0.14  0.13  0.13  0.13  0.13  0.12  0.12
##  [85]  0.12  0.12  0.11  0.11  0.11  0.11  0.11  0.11  0.11  0.10  0.10  0.10
##  [97]  0.10  0.10  0.09  0.09  0.09  0.09  0.09  0.09  0.09  0.08  0.08  0.08
## [109]  0.08  0.08  0.07  0.07  0.07  0.07  0.07  0.07  0.07  0.06  0.06  0.06
## [121]  0.06  0.06  0.05  0.05  0.05  0.04  0.04  0.03  0.00  0.00  0.00  0.00
## [133]  0.00
```

This code chunk performs PCA on the input data and reports the percentage of variance explained by the PCs. When the first few PCs explain a bulk of the total variation in the data, it is a good sign. Let us project the samples onto the first two PCs.

```
ccc = rep("gray", nsamples)
ccc[sdata$DM == 1] = "red"
```

```r
Thres = max(abs(tmp.pca$x[,1])) / 2  ### automatically calculate the axis range
XLAB = paste("PC1 (", vv[1], "%)", sep="")  ## PC1
YLAB = paste("PC2 (", vv[2], "%)", sep="")  ## PC2
library(scales)  ### To allow opaqueness in dots

#pdf("PCAplot.pdf", height=5.5, width=5, useDingbats = FALSE)
plot(tmp.pca$x[,1], tmp.pca$x[,2],   ### X=PC1, Y=PC2 coordinate
     col=alpha(ccc,0.2), pch=19,
     xlim=c(-Thres,Thres), ylim=c(-Thres,Thres),   ### Range set equally for both axes
     xlab=XLAB, ylab=YLAB,   ### Automatically assembled above
     main="MEC cohort (N=2,299)", cex=0.8)
legend("bottomright", c("No event","Incident DM"), pch=19, col=c("gray","red"), cex=0.8)
abline(v=0, lty=2)
abline(h=0, lty=2)
```

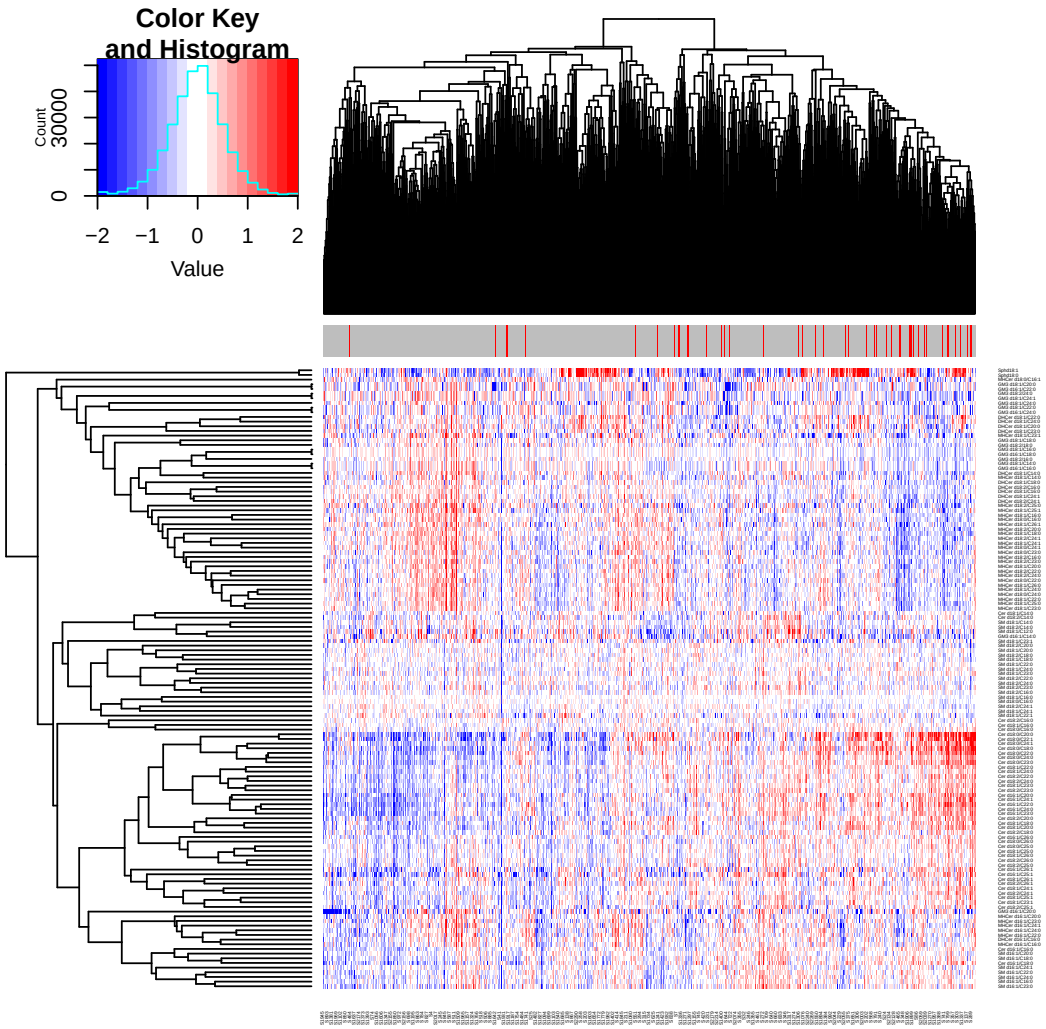



```
#dev.off()
```

This plot says that the incident DM cases are enriched in the bottom of the projection plot, i.e. along the PC2 axis. The observation suggests that the lipids with large contribution to the definition of PC2 may have predictive value for DM incidence. We can visualize the pattern with a heatmap:

```
################ Draw the entire data (on a relative scale)
tmp.ctr = sweep(tmp, 2, apply(tmp, 2, median))
### We are normalizing each lipid by its own median value
```

```
### so that all lipids are on a comparable scale.

#install.packages("gplots")
library(gplots)

##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##    lowess

#pdf("heatmap.pdf", height=20, width=25, useDingbats = FALSE)
heatmap.2(as.matrix(t(tmp.ctr)), trace="n", col=bluered(20), breaks=seq(-2,2,by=0.2),
          distfun=function(x) as.dist(1-cor(t(x))),
          hclustfun=function(x) hclust(x, method="average"),
          ColSideColors = ccc,
          cexRow=0.2, cexCol=0.2,
          mar=c(10,10))
```



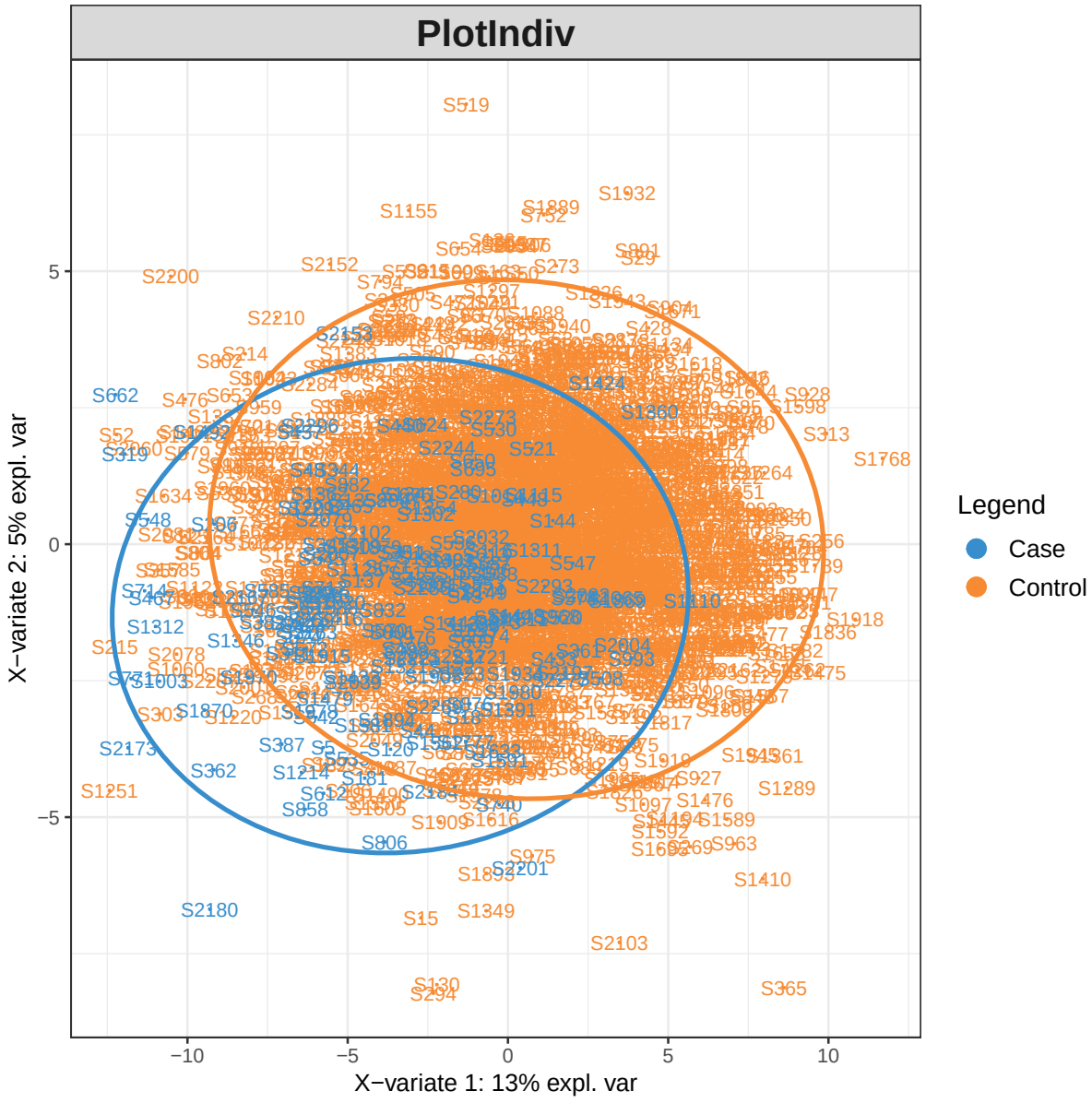

```
#dev.off()
```

The heatmap confirms two things: (i) Incident diabetes cases are enriched on the right hand side of the heatmap; (ii) ceramides and sphingomyelins are higher and hexosyl ceramides are lower at baseline in high-risk indiduals.

In metabolomics studies, many papers also show a projection plot called "PLS-DA" or "OPLS-DA", variants of partial least squares - discriminant analysis. Unlike PCA, this is a supervised analysis in the sense that the axes on which the samples are shown are found in a way that shows the largest contrast between the sample groups. In PCA, we did not use which participants are DM cases and which were

not. By contrast, PLS-DA explicitly uses this information. Here is the code snippet:

```
############### PLS-DA analysis (for fun)
############### Explain why PLS-DA is a "supervised" analysis
#if (!requireNamespace("BiocManager", quietly = TRUE))
#  install.packages("BiocManager")
#
#BiocManager::install("mixOmics")

library(mixOmics)

## Loading required package: MASS
## Loading required package: lattice
## Loading required package: ggplot2
##
## Loaded mixOmics 6.14.0
## Thank you for using mixOmics!
## Tutorials: http://mixomics.org
## Bookdown vignette: https://mixomicsteam.github.io/Bookdown
## Questions, issues: Follow the prompts at http://mixomics.org/contact-us
## Cite us:  citation('mixOmics')

sample.class = ifelse(sdata$DM == 1, "Case", "Control")
X = as.matrix(tmp)
Y = sample.class
tmp.out = plsda(X=X, Y=Y, ncomp=2)
#pdf("PLSDA_projection.pdf")
plotIndiv(tmp.out, ind.names = TRUE, ellipse = TRUE, legend = TRUE)
```




```
#dev.off()
```

3 Understanding the correlation structure between lipids and other risk factors

So far we analyzed the lipidomics data on its own, at most in association with the clinical endpoint (DM incidence). That is, we did not probe their correlation with other clinical parameters in the causal pathway. In the sample meta data,

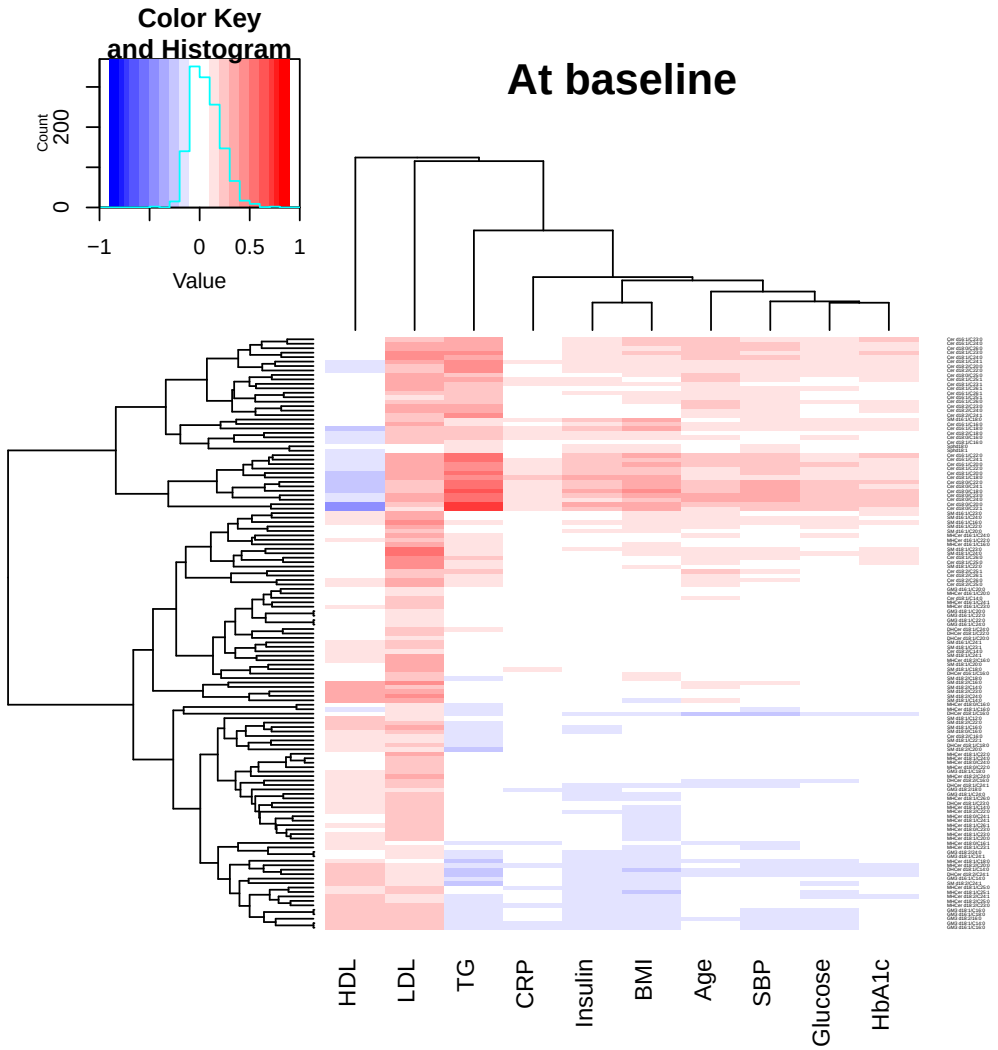
```
head(sdata)
```

```
##   ID DM Days Age Gender   BMI HbA1c SBP  HDL  LDL  CRP   TG Insulin Glucose
## 1 S1  0   NA  52      2 20.45    NA 105 1.76 4.16 0.29 1.12    4.65    4.43
## 2 S2  0   NA  54      1 22.93 5.558 137 1.64 2.74 0.72 0.90    5.87    4.30
## 3 S3  0   NA  52      1 24.47 5.267 111 1.44 2.91 0.26 2.68    6.78    4.24
## 4 S4  0   NA  53      1 24.21 6.062 143 1.71 2.36 1.30 1.20    3.96    5.22
## 5 S5  1 2190  65      2 29.23 4.987 150 1.63 2.95 3.85 1.65   12.35    5.72
## 6 S6  0   NA  68      2 22.42 5.609 177 1.26 2.66 0.19 0.95    3.79    4.92
```

Hence, let us look at the correlation structure between the biochem / clinical data and the lipid levels. For continous scaled measurements, it's surprisingly easy to do this:

```
### We first re-arrange the columns of the sample data matrix
### so that all continous variables are shifted to the right
sdata = sdata[,c(1:3,5,4,6:14)]
cp = colnames(sdata)[5:14]  ### Storing variable names

### Now we compute correlations between clinical data and lipids in a one liner
### Recall that ``tmp'' object holds lipidomic data in log2 scale
cormat = cor(sdata[,5:14], tmp, use="pairwise.complete.obs")

### Plot it in a heatmap
#pdf("cor_heatmap.pdf", height=20, width=6, useDingbats = FALSE)
heatmap.2(as.matrix(t(cormat)), trace="n", main="At baseline",
          col=bluered(20), breaks=seq(-1,1,by=0.1),
          #distfun=function(x) as.dist(1-cor(t(x))),
          hclustfun=function(x) hclust(x, method="average"),
          cexRow=0.2, cexCol=1,
          mar=c(10,10))
```

```
## Warning in image.default(z = matrix(z, ncol = 1), col = col, breaks = tmpbreaks, : unsorted
'breaks' will be sorted before use
```




```
#dev.off()
```

From the heatmap, we conclude the following. First, LDL levels are positively correlated with most lipids in this cohort. In contrast, HDL levels are positively correlated with hexosyl ceramides but negative correlated with ceramides. Total triglycerides shows the opposite trend. Blood pressure, insulin, glucose, and HbA1c levels are age dependent, and specifically assocaited with cerdmides.

4 Association analysis: logistic regression

We now begin the second last stage of the analysis: regression modeling of lipid levels at baseline with clinical endpoint (DM incidence) as outcome variable. The data comes from a prospective study – blood samples were drawn at baseline and the subjects were followed over time. This is an important point since the design allows for regression modeling in accordance with a potential causal pathway.

Our hypothesis is that DM incidence risk depends on circulating lipid levels, with the risk modulated by other prominent risk factors such as age, BMI, and SBP, etc. Therefore, we start with a base model consisting of all known risk factors. And we add each lipid to the model, and see if the lipid still has residual statistical association with incident DM risk, after accounting for those risk factors.

```
### Causal pathway: Lipid --> DM incidence,
### controlling for HbA1c at baseline, age, gender
### Start the model with no lipid, "base model"
### Then run through a for loop to test contribution of one lipid
sdata$Gender = factor(sdata$Gender, levels=c(1,2))
baseModel = glm(DM ~ Age + Gender + BMI + HbA1c + SBP + HDL + LDL + TG,
                family=binomial, data=sdata)
summary(baseModel)

##
## Call:
## glm(formula = DM ~ Age + Gender + BMI + HbA1c + SBP + HDL + LDL +
##     TG, family = binomial, data = sdata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6194  -0.3555  -0.2049  -0.1200   3.5031
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -20.750393   1.867394 -11.112  < 2e-16 ***
## Age           0.034744   0.010815   3.213  0.00132 **
## Gender2       0.627206   0.216344   2.899  0.00374 **
## BMI           0.133470   0.027403   4.871 1.11e-06 ***
## HbA1c         2.040225   0.281218   7.255 4.02e-13 ***
```

```
## SBP           0.017534   0.005336   3.286  0.00102 **
## HDL          -0.724902   0.368472  -1.967  0.04915 *
## LDL          -0.189623   0.123222  -1.539  0.12383
## TG            0.437113   0.140752   3.106  0.00190 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 980.61  on 1908  degrees of freedom
## Residual deviance: 735.71  on 1900  degrees of freedom
##   (390 observations deleted due to missingness)
## AIC: 753.71
##
## Number of Fisher Scoring iterations: 6
```

This base model suggests that most parameter contributes to the increased (log) odds of DM incidence, except HDL (LDL is not statistically significant), adjusting the effects of one another. Our next operation is to throw one lipid into the model to see if the lipid is still significantly associated with the odds.

To do this, we first standardize the data by subtracting the mean of log2 values and setting the standard deviation to 1 for each lipid. This standardization is performed to facilitate the interpretation of regression coefficients, as you will see later.

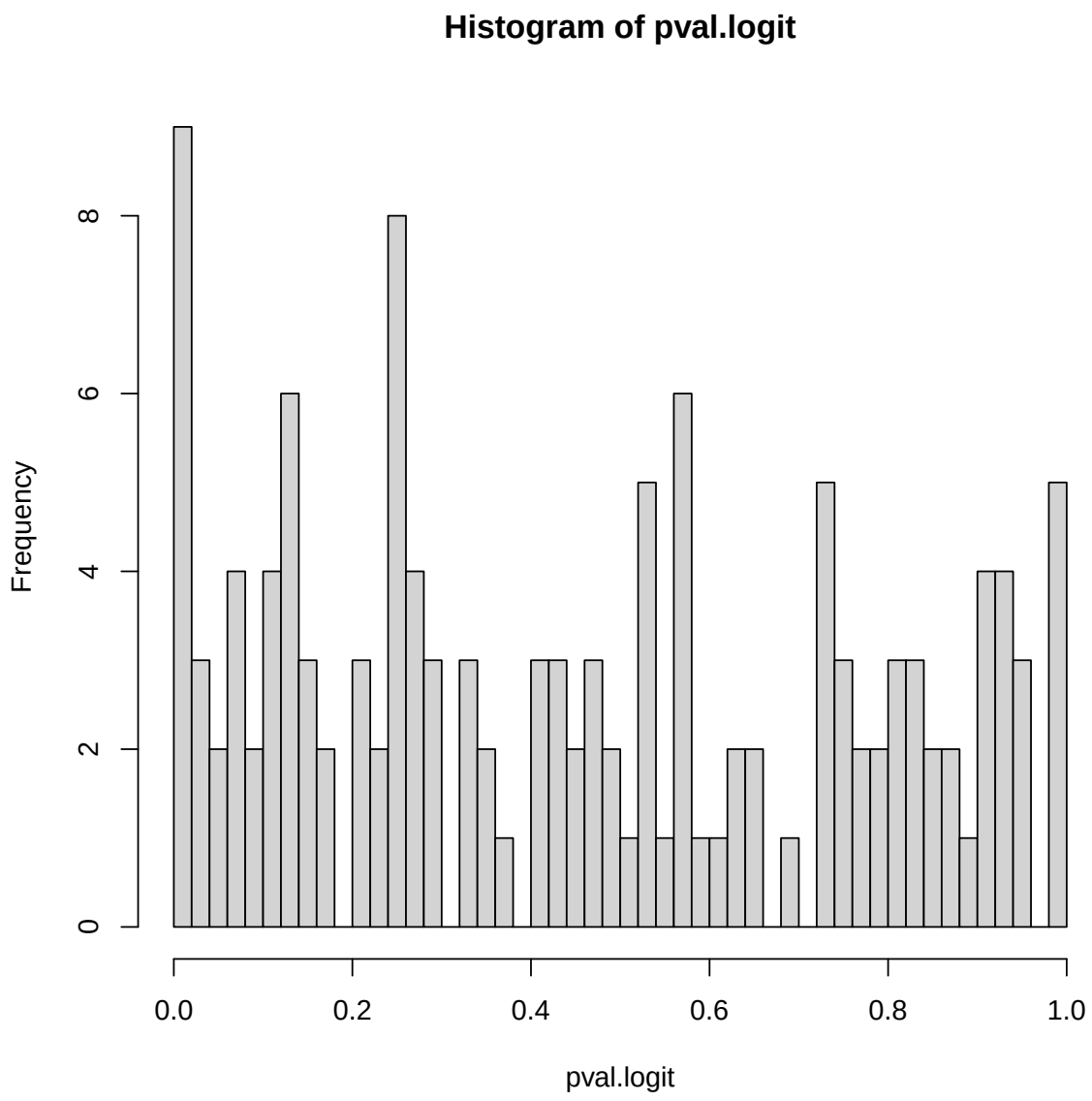
```
nlipid = ncol(tmp)
nsample = nrow(tmp)
lipid.name = colnames(tmp)
tmp2 = tmp

for(k in 1:nlipid) {
  mm = mean(tmp[,k], na.rm=TRUE)
  ss = sd(tmp[,k], na.rm=TRUE)
  tmp2[,k] = (tmp[,k] - mm) / ss
}
```

Once we have the data normalized, we next make placeholders for key statistics we're going to record for each lipid, and run through a for loop as follows:

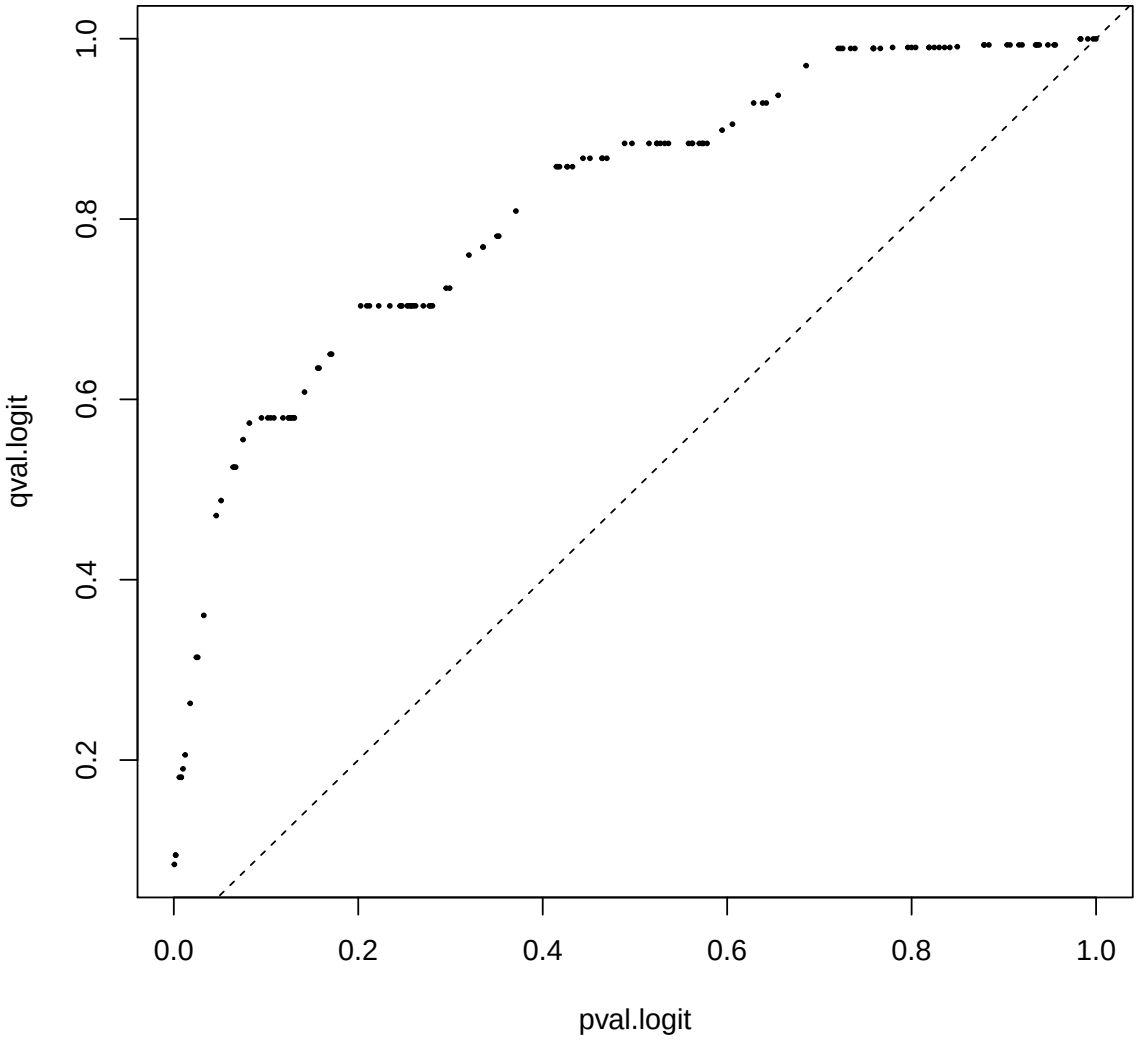
```r
### Build model (logistic)
coef.logit = rep(NA, nlipid)
pval.logit = rep(NA, nlipid)

for(k in 1:nlipid) {
  tmpModel = glm(DM ~ Age + Gender + BMI + HbA1c + SBP + HDL + LDL + TG + tmp2[,k],
                 family=binomial, data=sdata)
  coef.logit[k] = summary(tmpModel)$coef[10,1]
  pval.logit[k] = summary(tmpModel)$coef[10,4]
  if(k %% 50 == 0) print(k)
}
```

```
## [1] 50
## [1] 100
```

```r
hist(pval.logit, breaks=50)
```




```
# qvalue or Benjamini-Hochberg (BH)
#if (!requireNamespace("BiocManager", quietly = TRUE))
#  install.packages("BiocManager")
#BiocManager::install("qvalue")
library(qvalue)
qval.logit = qvalue(pval.logit)$qvalues
plot(pval.logit, qval.logit, cex=.3, pch=19)
abline(0,1,lty=2) ### properly modeled, it seems
```

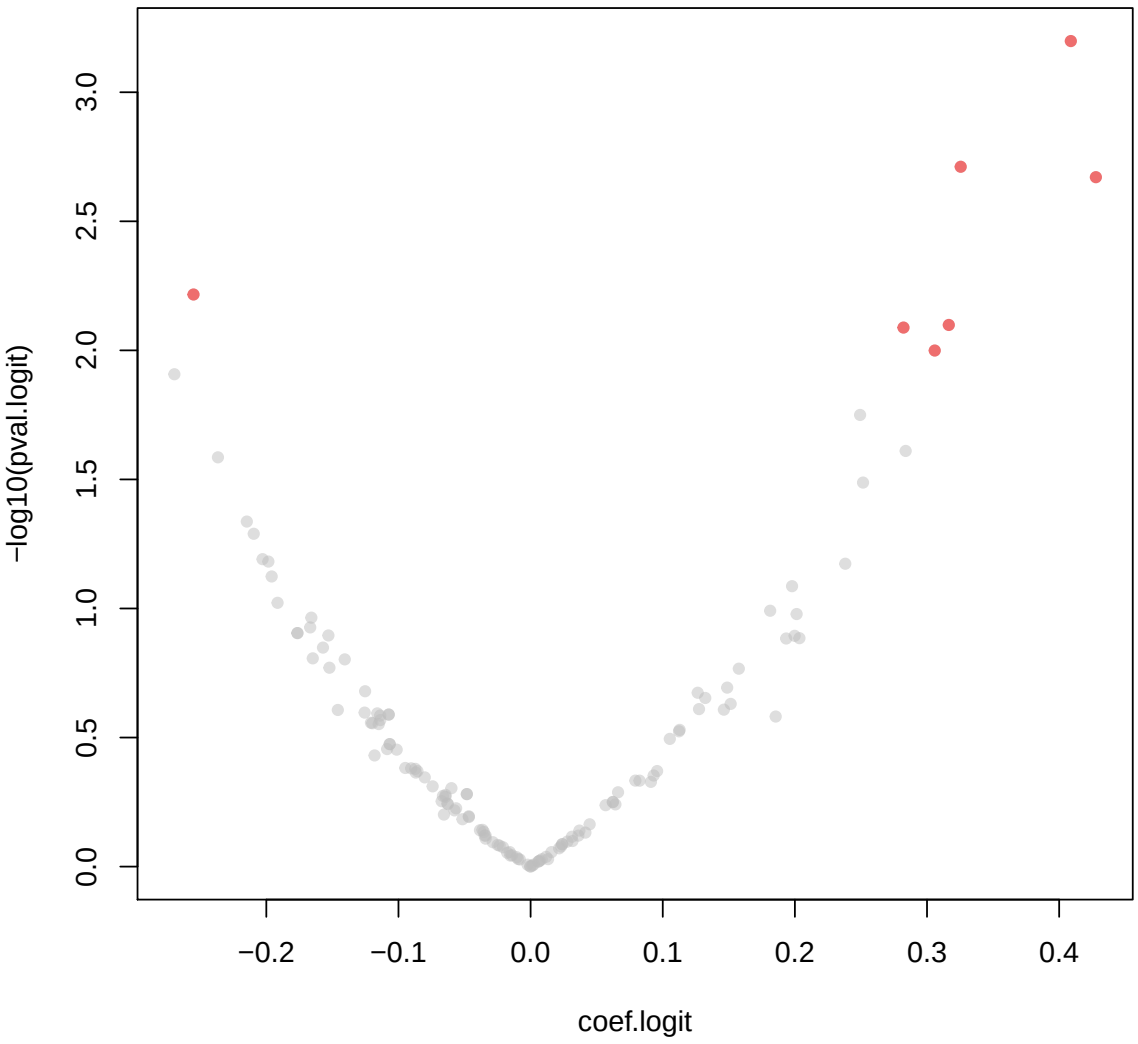



```
tab.logit = data.frame(lipid=lipid.name,
                       coefficient=round(coef.logit, 2),
                       pval=pval.logit, qval=qval.logit,
                       stringsAsFactors=FALSE, check.names=FALSE)
tab.logit = tab.logit[order(tab.logit$pval), ]
```

At this point, we have a table with statistical associations for lipids, with multiple testing correction (q-value). Since the total number of significant lipids is modest, we can be lenient in the FDR threshold. We draw a "volcano" plot with regression coefficients in the X-axis and -log10 p-value in the Y-axis, with

significant findings in red (FDR 20%).

```
# volcano plot with labels for significant ones
plot(coef.logit, -log10(pval.logit), pch=19, cex=.8, col=alpha("gray", 0.5))
sid = qval.logit <= 0.2
points(coef.logit[sid], -log10(pval.logit[sid]), pch=19, cex=.8, col=alpha("red", 0.5))
```




```
tab.logit[tab.logit$qval <= 0.2, ]

##                lipid coefficient         pval       qval
```

```
## 106    SM d16:1/C18:0        0.41 0.0006333953 0.08424157
## 116    SM d18:1/C18:0        0.33 0.0019439750 0.09456536
## 12    Cer d18:0/C18:0        0.43 0.0021330533 0.09456536
## 104 MHCer d18:2/C25:0       -0.26 0.0060745050 0.18086810
## 23    Cer d18:1/C18:0        0.32 0.0079711937 0.18086810
## 117    SM d18:1/C20:0        0.28 0.0081594632 0.18086810
## 2     Cer d16:1/C18:0        0.31 0.0100143746 0.19027312
```

This table contains the lipids with residual statistical significance after adjusting for known risk factors. However, this does not imply that these lipids have predictive properties: association is a weaker condition than predictiveness. Going into predictive analysis probably takes another tutorial session...

5 Association analysis with Cox regression (time-to-event)

The logistic regression analysis classifies each subject into two categories: incident DM case, or control (event free). In prospective studies, they often record the "time-to-event", i.e. the time from baseline to the time of DM diagnosis. For those subjects with no event recorded, time information refer to the time till last follow-up. In many cases, study participants are lost during follow-up for various reasons. This is called censoring. In our example data, however, we were not able to calculate the time to last follow-up accurately due to technical reasons. Therefore, we will add artificially large number of time for the event free subjects, pretending that all of them were followed through the entire study period. Even if we do this, the association model with time information often gives equivalent or better sensitivity of detecting true associations.

```
### Step 2: Cox model with time info (time-to-event)
#install.packages("survival")
library(survival)
mid = is.na(sdata$Days)
sdata$Days[mid] = 10000    ### Artificial follow-up date for event-free subjects
baseModelCox = coxph(Surv(Days, DM) ~ Age + Gender + BMI + HbA1c + SBP + HDL + LDL + TG,
                     data=sdata)
summary(baseModelCox)

## Call:
## coxph(formula = Surv(Days, DM) ~ Age + Gender + BMI + HbA1c +
##     SBP + HDL + LDL + TG, data = sdata)
```

```
##
##   n= 1909, number of events= 136
##    (390 observations deleted due to missingness)
##
##              coef exp(coef)  se(coef)      z Pr(>|z|)
## Age      0.033930  1.034512  0.009768  3.474 0.000514 ***
## Gender2  0.553661  1.739609  0.188266  2.941 0.003273 **
## BMI      0.119923  1.127411  0.022433  5.346 8.99e-08 ***
## HbA1c    1.845945  6.334082  0.241374  7.648 2.05e-14 ***
## SBP      0.016046  1.016176  0.004630  3.465 0.000529 ***
## HDL     -0.768789  0.463574  0.334563 -2.298 0.021568 *
## LDL     -0.184168  0.831796  0.108583 -1.696 0.089865 .
## TG       0.320409  1.377691  0.117168  2.735 0.006245 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##         exp(coef) exp(-coef) lower .95 upper .95
## Age        1.0345     0.9666    1.0149    1.0545
## Gender2    1.7396     0.5748    1.2028    2.5160
## BMI        1.1274     0.8870    1.0789    1.1781
## HbA1c      6.3341     0.1579    3.9466   10.1658
## SBP        1.0162     0.9841    1.0070    1.0254
## HDL        0.4636     2.1572    0.2406    0.8931
## LDL        0.8318     1.2022    0.6723    1.0291
## TG         1.3777     0.7259    1.0950    1.7333
##
## Concordance= 0.84  (se = 0.016 )
## Likelihood ratio test= 250.3  on 8 df,   p=<2e-16
## Wald test            = 232.6  on 8 df,   p=<2e-16
## Score (logrank) test = 265.7  on 8 df,   p=<2e-16
```

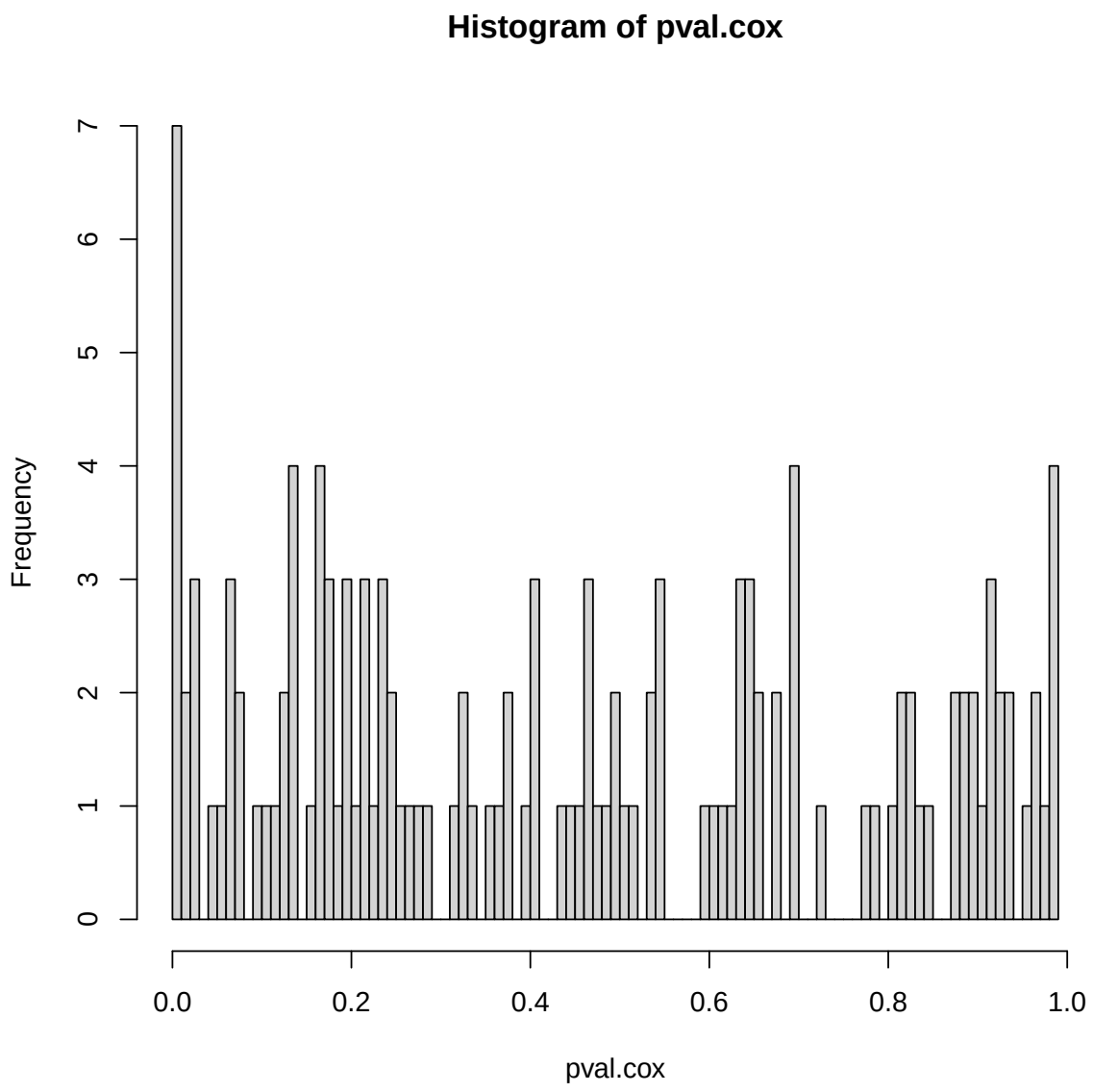
If you compare the base Cox model to the base logistic model fitted above, you will see that the association patterns (direction of regression coefficients, statistical significance scores) are quite consistent. If the follow-up times were properly recorded and there were many loss-of-follow-up cases, this consistency would have been shaken (possibly).

Now, we proceed to throw one lipid at a time onto this Cox model, by running through a for loop as before:

```
coef.cox = rep(NA, nlipid)
pval.cox = rep(NA, nlipid)

for(k in 1:nlipid) {
  tmpModel = coxph(Surv(Days, DM) ~ Age + Gender + BMI + HbA1c + SBP + HDL + LDL + TG + tmp2[,k],
                   data=sdata)
  coef.cox[k] = summary(tmpModel)$coef[9,1]  ### In Cox, they don't report intercept in summary()
  pval.cox[k] = summary(tmpModel)$coef[9,5]  ### Also, need to track the columns
  if(k %% 50 == 0) print(k)
}
```

```
## [1] 50
## [1] 100
```

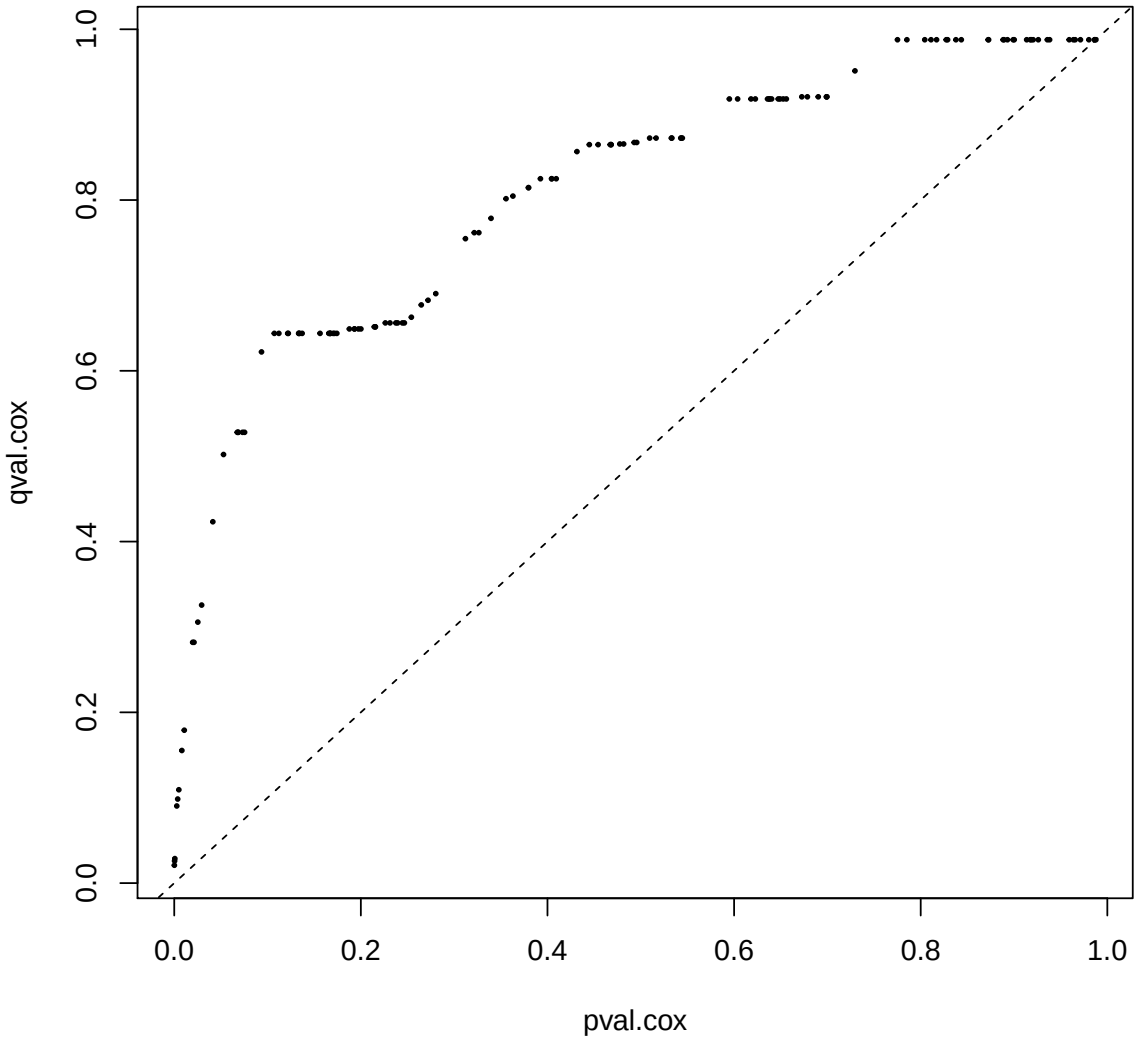
```
hist(pval.cox, breaks=100)
```



And multiple testing correction by q-value:

```
# qvalue or BH
#if (!requireNamespace("BiocManager", quietly = TRUE))
#  install.packages("BiocManager")
#BiocManager::install("qvalue")
library(qvalue)
qval.cox = qvalue(pval.cox)$qvalues
plot(pval.cox, qval.cox, cex=.3, pch=19)
```

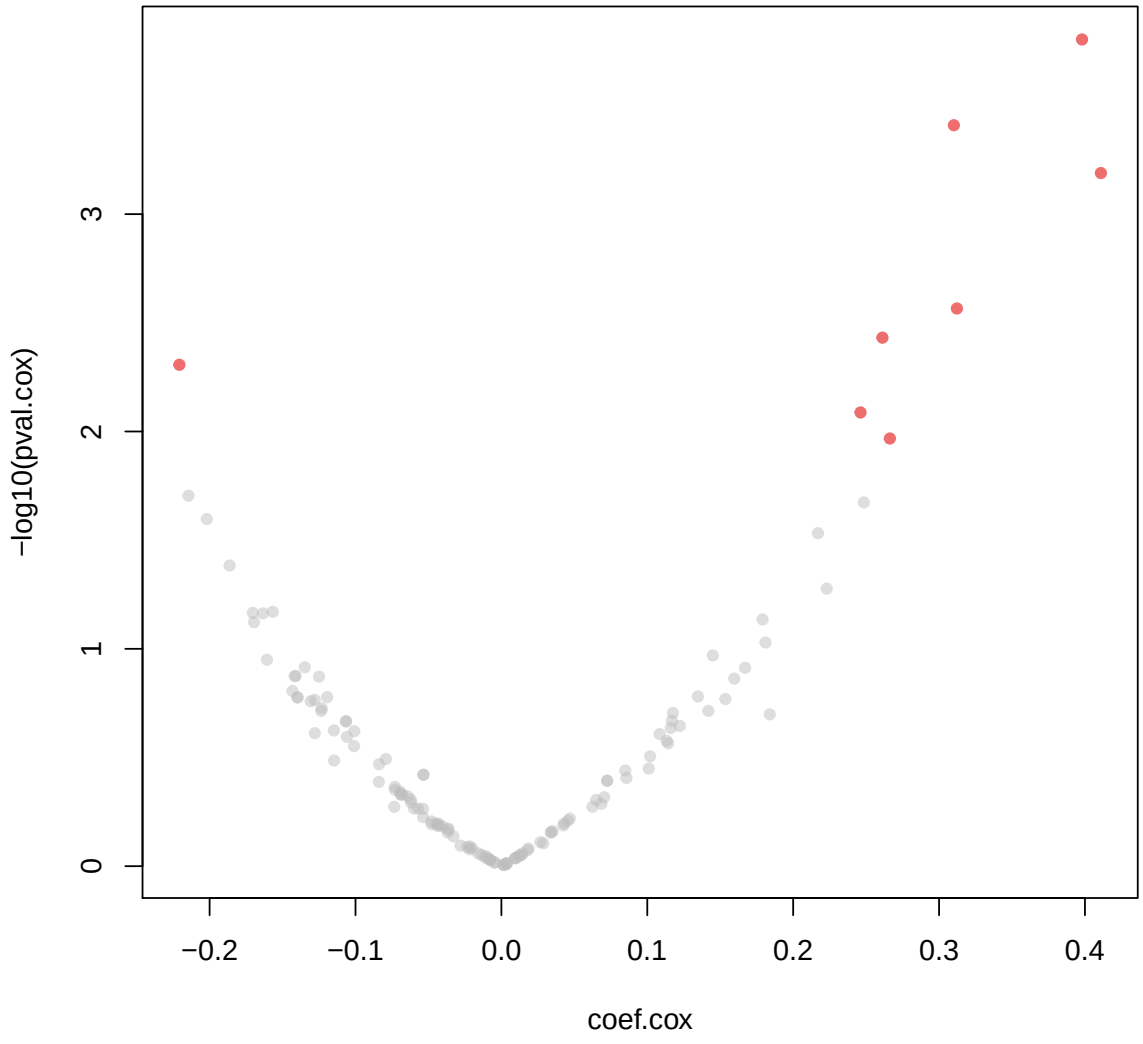
```
abline(0,1,lty=2) ### properly modeled, it seems
```




```
tab.cox = data.frame(lipid=lipid.name, coefficient=round(coef.cox, 2),
                     pval=pval.cox, qval=qval.cox,
                     stringsAsFactors=FALSE, check.names=FALSE)
tab.cox = tab.cox[order(tab.cox$pval), ]
```

Again, we see that the results for statistical significance (FDR 20%) are almost identical to that of the logistic regression analysis above.

```
# volcano plot with labels for significant ones
plot(coef.cox, -log10(pval.cox), pch=19, cex=.8, col=alpha("gray", 0.5))
sid = (qval.cox <= 0.2)
points(coef.cox[sid], -log10(pval.cox[sid]), pch=19, cex=.8, col=alpha("red", 0.5))
```



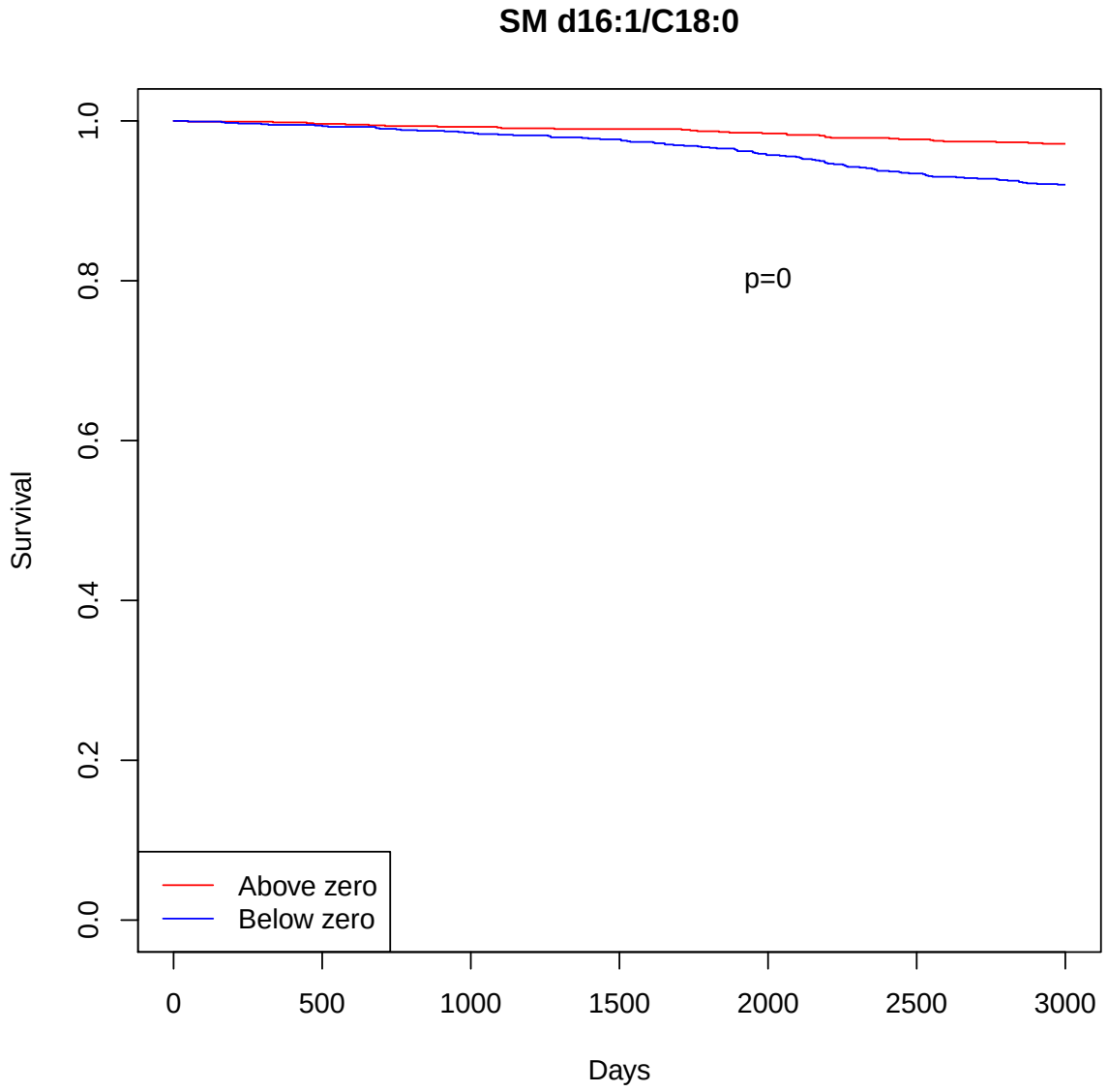

```
tab.cox[tab.cox$qval <= 0.2, ]
```

```
##                 lipid coefficient         pval       qval
## 106    SM d16:1/C18:0        0.40 0.0001571633 0.02090271
## 116    SM d18:1/C18:0        0.31 0.0003898359 0.02592409
```

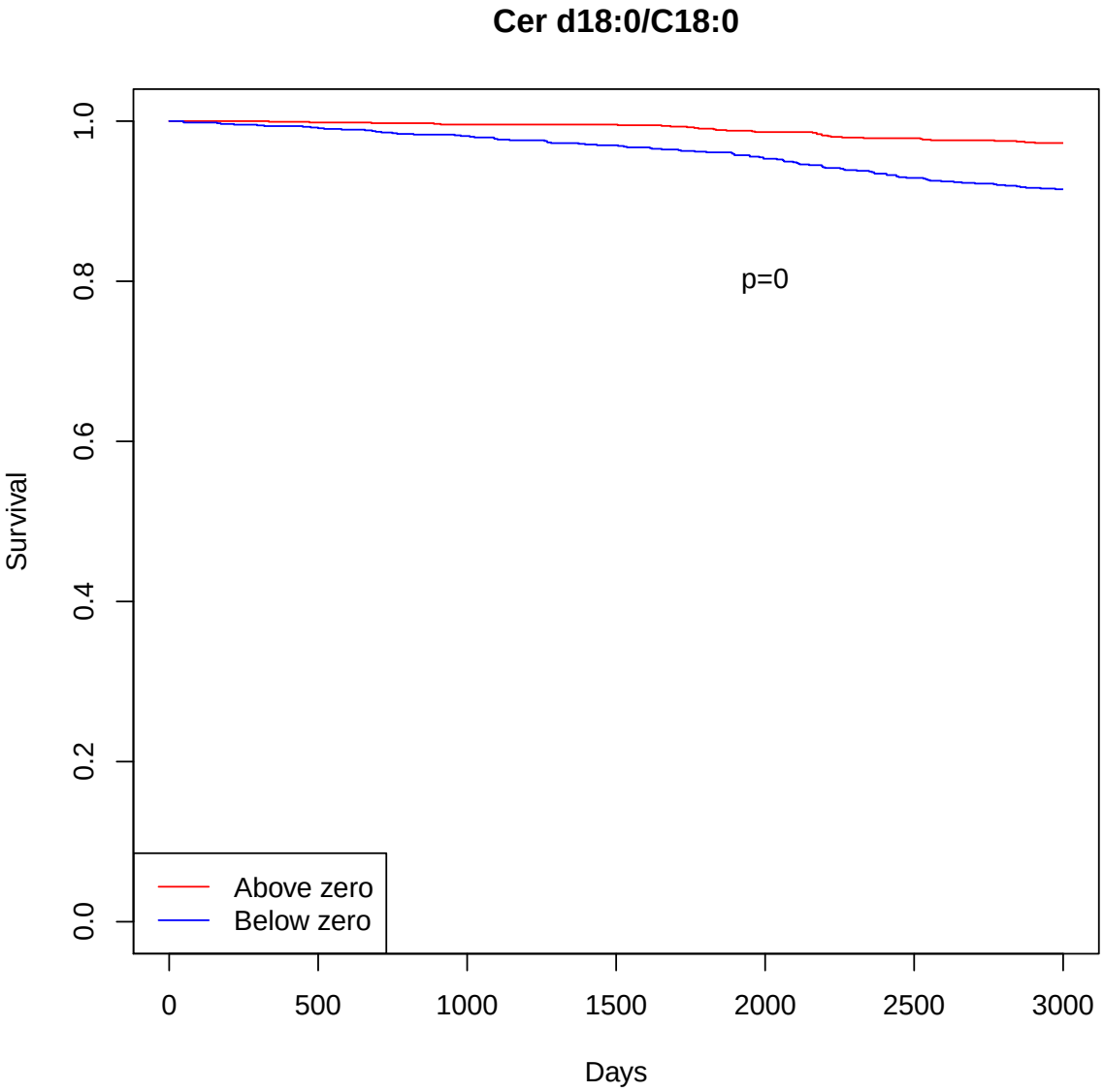
```
## 12    Cer d18:0/C18:0        0.41 0.0006470032 0.02868381
## 23    Cer d18:1/C18:0        0.31 0.0027163872 0.09031987
## 117    SM d18:1/C20:0        0.26 0.0037006416 0.09843707
## 104 MHCer d18:2/C25:0       -0.22 0.0049314165 0.10931306
## 49  DHCer d18:1/C18:0        0.25 0.0081742495 0.15531074
## 2     Cer d16:1/C18:0        0.27 0.0107700975 0.17905287
```

One way to visually verify the impact of a molecule selecte by Cox regression on the DM risk is to draw Kaplan-Meier curves in percentile groups. For simplicity, we look at above and below mean value for each lipid. See the code below:

```
##### Kaplan-Meier curve
sdata$`SM d16:1/C18:0` = tmp2$`SM d16:1/C18:0`
km1 = survfit(Surv(Days, DM) ~ (`SM d16:1/C18:0` > 0), data=sdata)
km1.diff = survdiff(Surv(Days, DM) ~ (`SM d16:1/C18:0` > 0), data=sdata)

sdata$`Cer d18:0/C18:0` = tmp2$`Cer d18:0/C18:0`
km2 = survfit(Surv(Days, DM) ~ (`Cer d18:0/C18:0` > 0), data=sdata)
km2.diff = survdiff(Surv(Days, DM) ~ (`Cer d18:0/C18:0` > 0), data=sdata)

plot(km1, lty=1, col=c("red","blue"), mark.time=TRUE,
     xlab="Days", ylab="Survival", main="SM d16:1/C18:0", xlim=c(0,3000))
legend("bottomleft", c("Above zero", "Below zero"), lty=1, col=c("red","blue"))
p.val.1 <- 1 - pchisq(km1.diff$chisq, length(km1.diff$n) - 1)
text(2000, 0.8, paste("p=", round(p.val.1, 4), sep=""))
```




```
plot(km2, lty=1, col=c("red","blue"), mark.time=TRUE,
     xlab="Days", ylab="Survival", main="Cer d18:0/C18:0", xlim=c(0,3000))
legend("bottomleft", c("Above zero", "Below zero"), lty=1, col=c("red","blue"))
p.val.2 <- 1 - pchisq(km2.diff$chisq, length(km2.diff$n) - 1)
text(2000, 0.8, paste("p=", round(p.val.2, 4), sep=""))
```



While the sphingolipid species seems to segregate the two groups in the later stage of the follow-up (5 years and beyond), the ceramide species can separate the risk profiles from the early period.